Applied Mathematical Sciences, Vol. 11, 2017, no. 10, 445 - 460
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ams.2017.7116

A New Reversible Watermarking Method Based on Histogram Shifting

Kim Sao Nguyen

University of Transport and Communication
Ha Noi, Viet Nam

Quang Hoa Le

Ha Noi University of Science and Technology
Ha Noi, Viet Nam

Van At Pham

University of Transport and Communication
Ha Noi, Viet Nam

Abstract

This paper proposes a new histogram shifting-based reversible watermarking method. In contrast to other almost methods that need to use simultaneously the watermarked image and some image-dependent side information to restore the watermark and the original image, our method can do above restorations without any auxiliary information. Such methods are called free from side information. Our solution is combining LSB-embedding and histogram shifting (HS). The original image is parted into two domains: the first one, consisting of a few pixels, is used for hiding auxiliary information by LSB-inserting, while the second one, comprising the other pixels is used for embedding the watermark by histogram shifting. In comparison to a few known free from side information methods, our method has higher embedding capacity and lower computational complexity. In addition, our solution can easily be used to make almost other histogram shifting-based reversible watermarking methods become free from side information.

Keywords: histogram shifting-based reversible watermarking; reversible watermarking

1 Introduction

Watermarking is to embed some data (called watermark) in digital images to authenticate and protect copyright of image products.

Traditional watermarking can only extract watermark but can not restore the original image. However, for many applications such as military, education and healthcare, restoring original image is compulsory. Therefore, reversible watermarking research is increasingly interested. Reversible watermarking is data hiding technique that is able to simultaneously extract watermark and to recover the original image.

The main research directions of reversible watermarking so far include: lossless compression, difference expansion, histogram shifting (abbreviated as HS), integer transformations, using JPEG characteristics, modulo, predictions

Reversible watermarking based on lossless compression [4] performs compression of lower bit sequence of pixels in order to make empty space, then use this space to embed watermark

In methods based on integer transforms such as contrast mapping [2], integer wavelet transformation [15] first a reversible integer transform is applied to image space, then information is embedded in transformed domain and original image is restored by using the inverse transform.

Difference expansion [13] is an effective method and interested by many researches. Difference between two consecutive points is expanded to the left side and one watermark bit is embedded in the right of difference.

In methods using jpeg specifics [1, 3, 7], quantified DCT blocks are usually used to embed watermarked bits.

In method Modulo 256 plus [16], watermarked image is made by adding modulo 256 between original image and watermark.

Recently, there are watermarking methods based on prediction-error matrix [5] that are constructed by differencing between original image and it's prediction one.

According to [11], important criteria to compare and evaluate the reversible watermarking methods are embedding capacity, image quality and computational complexity. If the difference extension methods [13] have higher embedding capacitiy, then histogram shifting methods [5, 6, 8, 10, 12] have better image quality.

The reversible watermarking methods based on histogram shifting is usually done as follows: first shifting the histogram of the original image to create a consecutive pair of two image pixels a and b that $h(a)$ is big and $h(b)$ equals 0 ($h(x)$ is the histogram of the image at point x). Then embedding an sequence of $h(a)$ bits on the pixels with value a.

These methods give good image quality, because the pixels only change one unit at most. However, to restore watermark and original image we need to know a, while this value is very hard to integrate (package) in the watermarked image. Between the sender and the receiver, therefore, there is must be an additional information (such as value a) transfer. In addition, without being able to pack all necessary

information into watermarked image also leads to other restrictions, such as can not apply multi-leveled watermark and construct public-key watermarking schemes.
Recently there are a few of methods that can overcome the weaknesses above: Hwang and partners [6] (now called Hwang method) selected a special pair of values a, b, such that this values remain unchanged after embedding the watermark. Thus, a can be defined from the watermarked image. Pan et al [12] (now called Pan method) employ a idea similar to one of [6] that will be explained in section III below. Fujiyoshi Masaaki [5] (now called MF method) use the same method in [10] to define a, b and then find the relationship between the original image and the watermarked image at a, b points. Then value a is determined by tracking the pixels satisfying above relationship.
Both of these methods have one big disadvantage: it takes a lot of time to find the value a and the embedding capacity isn't high.
In this paper, we'll present a new solution to integrate a into the watermarked image by dividing the original image into a small domain (8 pixels) and the rest domain (large domain). The values a, b is determined by histogram shifting technique on large domain (not on the entire image). Value a (including 8 bits) is embedded into the small domain by least significant bit (LSB) insertion technique. In this way, determining a from the watermarked image very simple and fast. This solution can be easily applied to improve almost known histogram shifting watermarking methods to pack products.
The methods that don't need to use side information to restore the original image and watermark, are called in [5] free from side information, here we call them as integrating or packaging methods.
By both theory analysis and experiments, it is proved that the proposed method has lower computation complexity and higher capacity than MF and Hwang methods.
Next content of the paper is organized as follows: section 2 presents the base some knowledge about the reversible histogram shifting watermarking. Section 3 introduces some related works including MF and Hwang methods. Section 4 presents proposed method. Section 5 does the some comparison between the proposed method with related methods by theory analysis and the computer experiments. Finally the conclusion is made in section 6.

2 The knowledge base

This paper only considers scale gray images which have pixel value in domain $\mathfrak{D}$:

$$\mathfrak{D} = \{x \in \mathbb{Z} \,|\, 0 \leq x \leq 255\}$$

A gray scale image I of size $M \times N$ can be considered as a $M \times N$ matrix (M rows, N columns) including elements $I(i,j)$ of $\mathfrak{D}$. Sometimes we just consider the subdomain J of I, and denote $P(J)$ as set of pairs (i,j) of J. In other word:

$$P(I) = \{(i,j) | 1 \leq i \leq M, 1 \leq j \leq N\}, P(J) \subseteq P(I)$$

Now we present the concept of histogram, the concept of histogram pair and methods based on histogram shifting for image I.

2.1 Histogram and histogram pair

Histogram of the image I (or domain J) at point $x \in \mathfrak{D}$, symbol $h(x)$, is the number of pixels in I (or in J) with value x.

According to [15], two consecutive values a, b ($b = a + 1$ or $a - 1$) on domain $\mathfrak{D}$ is called a histogram pair if:

$$h(a) > 0, h(b) = 0$$

Below, a and $h(a)$ are called vertex and hight of the pair (a, b), respectively.

Each pixel $(i, j) \in P(I)$ with value $I(i, j) = a$ can embed a bit w as follow:

$$I'(i,j) = \begin{cases} a \text{ } if \text{ } w = 0 \\ b \text{ } if \text{ } w = 1 \end{cases}$$

Then, algorithm to restore w and $I(i, j)$ from $I'(i, j)$ is simple as follows:

$I(i, j) = a$,
if $I'(i, j) = a$ then $w = 0$ else $w = 1$.

Remark 2.1: By using histogram pair (a, b), we can embed reversible a sequence $h(a)$ bit on pixels (i, j) with value $I(i, j) = a$.

Remark 2.2: After embedding, number of pixels with value a decreased by about a half (assuming the bit 0 and 1 in watermark approximately equal), so $h(a)$ is decreased about 2 times. In other words:

$$h'(a) \approx \frac{1}{2} h(a)$$

in which, h' is histogram of watermarked image I'

2.2 Histogram Shifting

Assuming pixel a has $h(a) > 0$. To make pair $(a, a - 1)$ become a histogram pair we can use histogram shifting technique as follows: First find point $z < a$ with $h(z) = 0$, then shift the histogram on $[z + 1, a - 1]$ to the left as follows:

```
for (i,j) ∈ P(I) and I(i,j) ∈ [z + 1, a − 1]
  I'(i,j) = I(i,j) − 1
end
```

Similarly, to create right histogram pair $(a, a + 1)$ we can do like this: first find point $z > a$ with $h(z) = 0$, then shift histogram on $[a + 1, z - 1]$ to the right as follows:

```
for (i,j) ∈ P(I) and I(i,j) ∈ [a + 1, z − 1]
  I'(i,j) = I(i,j) + 1
end
```

Remark 2.3: Image after shifted histogram to the left can be easily restored by shifting histogram to the right and vice versa.

The reversible watermarking methods based on histogram shifting

These methods first use a HS technique to create histogram pairs, then embed the watermark on the received ones.

To create a pair of histogram, according to the section 2.2, we need to find point z in which $h(z) = 0$. If point z like that does not exist, we can select $z < a$ (or $z > a$) with smallest $h(z)$, they often symbolized $minL$ (or $minR$). Then, to restore the image, we need to add value $minL$ $(or\ minR)$ and position of the pixels $I(i,j)$ with value $minL$ $(or\ minR)$ in front of watermark to create an array of embedded bits. This array, denote by B, will include information H and watermark W:

$$B = H \oplus W$$

Methods based on HS differ mainly by ways to create a histogram pairs for improving embedding capacity. The group of this methods has the advantage of the watermarked image quality because image pixels only have to change maximum 1 value. To restore B and the original image we need to know the vertex of the histogram pairs. This information may be hardly to integrate into the watermarked image, so usually it is transferred outside. Therefore, if you only know the watermarked image, it is not enough to restore the necessary information. Most of the methods of watermarking suffer from this weakness.

Recently, there are few papers which can overcome the above disadvantage such as the MF and Hwang methods. However, these methods still have a number of restrictions on the capacities and computational complexity as shown below.

3 The related paper

3.1 Hwang Method

Hwang and partners, first find the value $peak$ which achieves maximum of histogram in $\mathfrak{D}$:

$$h(peak) = max\{h(x)|x \in \mathfrak{D}\}$$

(If there are more than one maximum point, select the first). Then the histogram is shifted to the left and to the right in order to create two pairs of histogram $(peak - 1, peak - 2)$ and $(peak + 1, peak + 2)$, and finally, use these pairs to embed watermark. By creating two special histogram pairs like above, the value $peak$ is unchanged after embedding. Therefore, it is possible to calculate this peak from the watermarked image I' with the formula:

$$h'(peak) = max\{h'(x)|x \in \mathfrak{D}\} \tag{3.1}$$

After that, the vertexes $peak - 1, peak + 1$ of the histogram pairs are defined and the watermark as well as the original image are restored.

Remark 3.1: Capacity of Hwang method on image I, symbol C_h, according to remark 2.1 is calculated by:

$$C_h = h(peak - 1) + h(peak + 1) \tag{3.2}$$

Remark 3.2: According to remark 2.2, if we continue to embed on I' (embedding level 2), capacity C'_h is just about half of C_h:

$$C'_h \approx \frac{1}{2} C_h \tag{3.3}$$

3.2 Pan Method

In order to extract the watermark and restore the original image without using any side information, Pan et al in [12] propose a method as follow: "select the peak point as the reference point and use the neighboring point of peak point to embed the secret bits". It is clearly that this method and Hwang method described in subsection 3.1 are absolutely similar. Therefore, two above methods have the same embedding capacity and computational complexity when using them for embedding the watermark on the entire image as well as on image's non-overlapped blocks. That is why, below we consider these two methods as one method

3.3 MF Method

MF method finds the value $peak$ of the histogram like first step of Hwang method. Then the histogram is shifted to the left to have a pair of histogram $(peak, peak-1)$ and embeds information on this pair. To find the value $peak$ from the watermarked image, M. Fujiyoshi uses the following property of the value $peak$:

$$h'(peak-1) + h'(peak) = h(peak) \tag{3.4}$$

The value $h(peak)$ is inserted into first 16 bits of H. In other words, $h(peak)$ is transformed into a 16-bit binary array and is embedded in the first 16 pixels of I that have value equals to $peak$, by using the algorithm in section 2.1.

Determining peak is performed by browsing each value x on all of domain $\mathfrak{D}$, with each x, restore first 16 bit of $I'(i,j)$ which value is equal x or $x-1$ following algorithm in section 2.1. Called $g(x)$ the restored value, if $h'(x-1)+h'(x)=g(x)$ then x satisfies the conditions (3.4), so this x is considered as the peak. From $peak$ found, it is easy to restore the original image as well as the watermark.

The MF method does not have high capacity because it only uses a histogram pair, and determining $peak$ by tracking each points as above costs a lot of time. Besides, there may have other values which also satisfy condition (3.4), it is uncertain that the value received from method above is the desired true $peak$.

Remark 3.3: The capacity of MF on I, symbols C_{mf} and the capacity on I', symbols C'_{mf}, are calculated by the formula:

$$C_{mf} = \max\{h(x) | x \in \mathfrak{D}\}$$
$$C'_{mf} = \max\{h'(x) | x \in \mathfrak{D}\}$$

Because the maximum height of h' approximately equals h, so C'_{mf} is close to C_{mf}.

4 Proposed method

The idea of proposed method is to divide the image I into two domains: I_1 includes 8 pixels and I_2 is the rest. Then build histogram $h(x)$ on I_2.

The embedding information is performed by HS on I_2 (not on I), the value $peak$ is stored by LSB on I_1. Concrete algorithm is described below:

4.1 Embedding Algorithm

Step 1: Divide the image into two domains I_1 and I_2. Here I_1 includes first 8 pixels of I, I_2 is the rest. However, theoretically, I_1 may include any 8 pixels of I. A random key can also be used to select the pixels for I_1.

Step 2: Build histogram $h(x)$ on I_2 and determine the values $peak, minL$ and $minR$ according to the formula:

$$h(peak) + h(peak + 1) = max\{h(x) + h(x + 1)|x \in \mathfrak{D}/\{255\}\}$$
$$h(minL) = min\{h(x)|x \in [0, peak], x \in \mathfrak{D}\}$$
$$h(minR) = min\{h(x)|x \in [peak, 255], x \in \mathfrak{D}\}$$

Step 3: determine the Supplemental information H:

Parameters	Meaning	Length (number of bits)
V	8 least significant bits of domain I_1	8
$minL$	Minimum Point on the left of $peak$	8
C_L	Number of pixels having value equal $MinL$	8
M_L	Set of positions of the pixels having value equal minL	$9 \times 2 \times C_L$
$minR$	Minimum Point on the right of $peak$	8
C_R	Number of pixels having value equal $MinR$	8
M_R	Set of positions of the pixels having value equal $minR$	$9 \times 2 \times C_R$

Step 4: determine the array of embedded bits.

$B = H \oplus W$

Here W is watermark which have length: $size(W) = h(peak) + h(peak + 1) - size(H)$

Step 5: embed value $peak$ by inserting LSB technique into the domain I_1.It results domain I'_1

Step 6: Create the histogram pair $(peak, peak - 1)$ and $(peak + 1, peak + 2)$ by shifting histogram as below:

```
For (i,j) ∈ P(I_2) and I(i,j) ∈ [minL + 1, peak − 1]
        I'(i,j) = I(i,j) − 1
End
```

And

```
For (i,j) ∈ P(I_2) and I(i,j) ∈ [peak + 2, minR − 1]
        I'(i,j) = I(i,j) + 1
End
```

Step 7: embed the binary sequence $B = b_1 b_2 \dots b_L$ ($L = size(B)$) on the pixels having value equal to $peak$ or $peak + 1$ as follow:

```
k = 0,
for (i,j) ∈ P(I_2) and I(i,j) ∈ [peak, peak + 1]
        k = k + 1
        if b_k = 0 then
                I'(i,j) = I(i,j)
        Else if I(i,j) = peak then
                        I'(i,j) = I(i,j) − 1
                else
                        I'(i,j) = I(i,j) + 1
                end
        end
end
```

After step 7 is performed, we get I_2'. The watermarked image I' includes two domains I_1' and I_2'.

Remark 4.1: After embedding B, the changed pixels include: points in I_1; points in I_2 having value belongs to $[minL, minR]$.

4.2 Extraction algorithm

Watermark W and original image I are restored from I' by following steps:

Step 1: Divide the image I' into two domains I_1' and I_2' as in the embedding algorithm.

Step 2: Extract 8 LSB of domain I_1' to get the value $peak$

Step 3: Based on $peak$, restore B in domain I_2' as follow:

```
k = 0,
for (i,j) ∈ P(I_2) and I(i,j) ∈ [peak − 1, peak + 2]
         k = k + 1
         if I'(i,j) ∈ [peak, peak + 1] then
                b_k = 0
        else
                b_k=1
        end
end
```

Step 4: From B we easily receive supplemental information H and watermark W

Step 5: Use H to restore the original image as follow:

5.1: Restore I_1: Insert the value V into LSB of pixels in I_1'

5.2: Restore pixels in I_2 with values in $[minL, minR]$ (reverse histogram shifting)

for $(i,j) \in P(I_2')$ and $I'(i,j) \in [minL+1, minR-1]$
    if $I'(i,j) < peak$
        $I(i,j) = I'(i,j) + 1$
    else if $I'(i,j) > peak + 1$ then
        $I(i,j) = I'(i,j) - 1$
    end
end

5.3: Restore pixels in I_2 with value equal $minL$ or $minR$

for $(i,j) \in P(I_2')$ and $I'(i,j) \in \{minL, minR\}$
    if $(i,j) \in M_L \cup M_R$ then
        $I(i,j) = I'(i,j)$
    else if $I'(i,j) = minL$ then
        $I(i,j) = I'(i,j) + 1$
        else $I(i,j) = I'(i,j) - 1$ end
    end
end

4.3 Example

To demonstrate the proposed method, we present an example in which I is showed in figure 1, W is an array of 8 bits: 00001111

6	5	5	8	7	5	5	5	1	6
5	6	5	5	5	6	5	6	5	7
5	6	6	7	4	5	5	5	2	4
6	5	9	5	6	6	9	6	5	5
6	5	6	5	3	6	8	2	6	7
5	5	5	5	5	6	11	6	6	6
5	6	6	6	6	5	6	6	6	8
6	5	5	5	6	5	5	6	5	6
6	6	0	5	6	3	5	6	6	5

Fig. 1. Original image

6	4	4	8	6	5	4	5	1	6
5	6	5	5	5	6	5	6	5	8
5	6	6	8	3	5	5	5	1	3
6	5	10	5	6	6	10	6	5	5
6	5	6	5	2	6	9	1	6	8
5	5	5	5	5	6	11	6	6	6
5	6	6	6	6	5	6	6	6	9
6	5	5	5	6	5	5	6	5	6
6	6	0	5	6	2	5	6	6	5

Fig. 2. Insert LSB in I_1 and histogram shifting on I_2
(italic numbers are modified)

6	4	4	8	6	5	4	5	1	7
4	7	4	5	4	6	5	6	5	8
5	6	6	8	3	5	5	4	1	3
6	5	10	5	6	6	10	6	5	4
6	5	6	5	2	6	9	1	6	8
5	5	4	5	5	6	11	6	6	7
5	6	7	6	6	5	6	7	6	9
7	5	5	5	6	5	5	6	5	6
6	6	0	5	6	2	4	7	7	4

Fig. 3. Embedding watermark W.
(bold numbers are modified)

Step 1: I_1 includes first 8 pixels of I (8 cells of the gray), I_2 is the rest.

Step 2: Histogram $h(x)$ on I_2 and relevant quantities:

x	0	1	2	3	4	5	6	7	8	9	10	11	12	...	Ip
$h(x)$	1	1	2	2	2	32	34	3	2	2	0	1	0	...	0

$peak = 5$ (binary form is 00000101); $minL = 1$; $minR = 10$

Step 3: Supplemental information H:

Parameter	V	$minL$=1	C_L=1	M_L= { (1,9)}
Binary form	11110100	00000001	00000001	000000001000001001
$minR$=10	C_R=0	M_R Empty		
00001010	00000000			

Step 4: Array of embedded bits (length $h(5) + h(6) = 32 + 34 = 66$):

$$B = H \oplus W = 000000001000001001000010101 1\_$$
$$\_110100000000010000000100000000000001111$$

Step 5: embed value $peak$ (00000101) by inserting LSB into domain I_1. It results domain I_1':

6	4	4	8	6	5	4	5

Step 6: Create the histogram pairs $(5, 4)$ and $(6,7)$ by HS, this results matrix in figure 2.

Step 7: embed B on the pixels with value is 5 or 6, we get watermarked image like figure 3.

5 Compare the proposed method with Hwang method, Pan method and MF method

This section will compare the methods by both theory analysis and computer experiment.

5.1 Theory analysis

Capacity:

For the proposed method, embedding ability to embed C_p on I and C_p' on I' are calculated in the formulas:

$$C_p = max\{h(x) + h(x + 1)|x \in \mathfrak{D}\}$$
$$C_p' = max\{h'(x) + h'(x + 1)|x \in \mathfrak{D}\}$$

From there and remarks 3.1, 3.2 and 3.3, it leads to the conclusion:

a. Capacity of the proposed method in the original I (embedding level 1) is not much larger than the Hwang and roughly twice the MF method.

b. Capacity of proposed method in the watermarked image I' (embedding level 2) is about twice larger than both two methods Hwang and MF

This conclusion is consistent with the results of the experiment in subsection 5.2.

The calculation complexity:

Three methods only differ in determining the value $peak$ of the watermarked image, so just need to focus on evaluating computational amount to find $peak$ in each method.

- Methods Hwang and Pan method: To determine $peak$ by the formula (3.1), first use $M \times N$ additions to build histogram h' of the image I' (size $M \times N$). Then, perform 256 comparison-operations to determine maximum of $h'(x)$. So, the amount of computation to determine $peak$ of Hwang include: $M \times N$ sums and 256 comparisons.

- Method MF: in this method, we also need to use $M \times N$ additions to build histogram $h'(x)$. Then, with each x of $\mathfrak{D}$, to check conditions:

$$h'(x-1) + h'(x) = g(x) \tag{5.1}$$

To determine 16 bits of $g(x)$ we need to browse the image I' (size $M \times N$) to find out 16 elements with the value equal x or $x-1$ (if x take bit 0, if $x-1$ take bit 1). This need $(16 + M \times N)/2$ comparisons in average. Then we need 15 shift-operations and 15 addition to transform 16 bits to decimal. So to determine $g(x)$ we need to do: $(16 + M \times N)/2$ comparisons, 15 shifts and 15 additions. After having $g(x)$, we need too one addition and one comparison to check condition (5.1). The x value should be browsed about $256/2$ times in average (half the elements of $\mathfrak{D}$). So the amount of computation to determine $peak$ in MF is about $(M \times N + 128 \times 16)$ additions, 128×15 shifts and $64 \times (18 + M \times N)$ comparisons.

- Methods proposed: value $peak$ is extracted from 8 pixels of I', so just only need to use: 8 bit-extracts, 7 shifts and 7 additions.

From the above analysis, a conclusion can be made: the computational amount of proposed method is much fewer than either at MF method and Hwang method. This is completely consistent with the results of the experimental in 5.2.

Quality:

Because embedding capacities of different methods are not the same, we cannot use PSNR standard to compare image quality. Objectively, here we use the distortion-rate of original image after embedding (call R_d - number of pixels to change to embed a bit) as a criteria to compare image quality. R_d is calculated in the formulas:

$$R_d = \frac{1}{L_W} \sum_{i=1}^{M} \sum_{j=1}^{N} |I(i,j) - I'(i,j)| \quad \text{with } L_W = size(W) \tag{5.2}$$

We realize, the image has been changed by shifting histogram. In addition, the number of pixels to change directly proportional with the number of embedded bits. So the distortion-rate of the three methods above is equivalent.

5.2 Experiment

To illustrate the results of theory analysis, we perform experiments on the sample image set in [17]. Images from 4 to 8 have size 512×512, image 9 has size of 256×256 and 10 has size 1024×1024. The watermark W is from the binary image 11. Programs are written in the language Matlab R2012a and ran on IdeaPad S410p Lenovo computer.

Fig. 4. Lena

Fig. 5. Peppers

Fig. 6. Sailboat

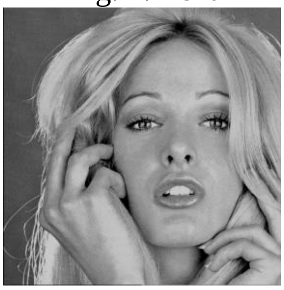

Fig. 7. Tiffany

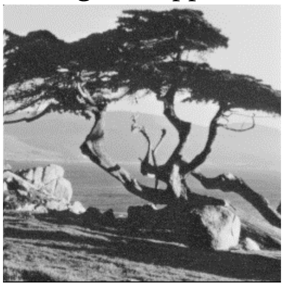

Fig. 8. Sight

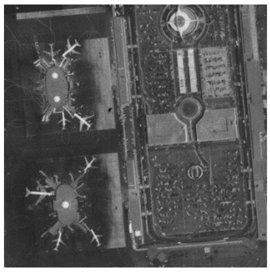

Fig. 9. Airport

Fig. 10. Boat

Fig. 11. Watermark

5.2.1 Compare capacities

Capacities on the image of each method on level 1 and level 2 is presented in two tables below.

Table 1: Capacities from the original image *I* (level 1)

STT	Image	Maximum number of embedded bits of methods		
		Hwang and Pan	MF	Proposed
1	Lena	5535	2919	5696
2	Pepper	5263	2712	5854
3	Tiffany	5788	2956	5854
4	Boat	7592	3906	7738
5	Sailboat	7109	3707	7364
6	Sight	2232	1200	2331
7	Airport	44894	22723	45395
	Total	78413	40123	80232

Table 2: Capacities on the watermarked image I' (level 2)

STT	Image	Maximum number of embedded bits of methods		
		Hwang and Pan	MF	Proposed
1	Lena	2737	2803	5421
2	Pepper	2628	2662	5243
3	Tiffany	2902	2934	5785
4	Boat	3847	3836	7556
5	Sailboat	3543	3657	6739
6	Sight	1096	1165	2298
7	Airport	22639	22673	44385
	Total	39392	39730	77427

Remark 5.1: From the tables above we can conclude: the capacity of proposed method

In level 1 is about 1.02 times more than Hwang and twice more than MF.

In level 2 is about 1.97 times more than Hwang and 1.95 times more than MF.

At both levels is about 1.34 times more than Hwang and 1.97 times more than MF.

5.2.2 Compare the time to determine $peak$ from watermarked image

The table below presents statistics time to determine peak from watermarked images of each methods on different original images.

Table 3: Time to determine $peak$

STT	Image	Time to determine peak value (Measured in micro seconds)		
		Hwang and Pan	MF	Proposed
1	Lena	6700	949500	7.5
2	Pepper	6600	491200	7.4
3	Tiffany	6600	1891400	7.5
4	Boat	6600	579900	7.5
5	Sailboat	6700	635700	7.5
6	Sight	1600	186100	7.5
7	Airport	29100	89700	7.5
	Total	63900	4823500	52.4

Remark 5.2: Table 3 shows that time to determine $peak$ of Hwang method and MF method depends on the size of the image, while proposed method's is independent.

Remark 5.3: time to determine $peak$ of Hwang method more than 1219 times and of the MF method more than 92051 times compared with the proposed method.

5.2.3 Compare the image distortion rate

Table 4: The image distortion rate

STT	Image	The image distortion rate		
		Hwang and Pan	MF	Proposed
1	Lena	46.73	65.68	45.91
2	Pepper	49.23	34.88	44.69
3	Tiffany	43.10	53.22	44.01
4	Boat	33.71	54.92	33.58
5	Sailboat	36.10	51.61	35.33
6	Sight	28.84	6.93	28.06
7	Airport	44.67	10.05	22.62
	Average	40.34	39.61	36.31

Table 4 shows that the image distortion rate of the methods are not much differ, but at average level, the proposed method has lower image distortion rate.

Conclusion

The paper suggests a free from side information watermarking method (packaged image) by combining inserting LSB and histogram shifting. From the watermarked image we can extract watermark and original image without any additional information outside. Proposed solution can be easily used for most known histogram shifting watermarking methods in order to pack the watermarked image.

To compare with known packaging methods, the proposed method has higher capacity and lower computational complexity.

References

[1] A. Almohammad, G. Ghinea, R. M. Hierons, JPEG steganography: A performance evaluation of quantization tables, *2009 International Conference on Advanced Information Networking and Applications*, (2009), 471-478. https://doi.org/10.1109/aina.2009.67

[2] D. Coltuc, J. M. Chassery, Very fast watermarking by reversible contrast mapping, *IEEE Signal Processing Letters*, **14** (2007), 255-258. https://doi.org/10.1109/lsp.2006.884895

[3] Van Tuan Do, Kim Sao Nguyen, Thanh Toan Nguyen, Van At Pham, A new data hiding method base on jpeg image, *Journal of Vietnam Ministry of Information and Communication,* (2014), 41-52.

[4] J. Fridrich, M. Goljan, R. Du, Invertible authentication, Proc. SPIE 4314, *Security and Watermarking of Multimedia Contents III,* (2001), 197-208. https://doi.org/10.1117/12.435400

[5] M. Fujiyoshi, A histogram shifting-based blind reversible data hiding method with a histogram peak estimator, *2012 International Symposium on Communications and Information Technologies (ISCIT),* (2012), 313-318. https://doi.org/10.1109/iscit.2012.6380913

[6] J. Hwang, J. Kim, J. Choi, A reversible watermarking based on histogram shifting, *2012 International Symposium on Communications and Information Technologies* (ISCIT), (2012), 348-361.

[7] M. Iwata, K. Miyake, A. Shiozaki, Digital steganography utilizing features of JPEG images, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* E87-A (4), (2004), 929-936.

[8] W.-C. Kuo, D.-J. Jiang, Y.-C. Huang, Reversible data hiding based on histogram, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, (2014), 1152-1161.

[9] A. Khan, A. Siddiqa, S. Munib, S. A. Malik, A recent survey of reversible watermarking techniques, *Information Sciences*, **279** (2014), 251-272. https://doi.org/10.1016/j.ins.2014.03.118

[10] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, Reversible data hiding, *IEEE Transactions on Circuits and Systems for Video Technology,* **16** (2006), no. 3, 354-362. https://doi.org/10.1109/tcsvt.2006.869964

[11] B. Ou, Y. Zhao, R. Ni, Reversible watermarking using prediction error histogram and blocking, Chapter in *Digital Watermarking*, 2011, 170-180. https://doi.org/10.1007/978-3-642-18405-5_14

[12] Z. Pan, S. Hu, X. Ma, L. Wang, Reversible data hiding based on local histogram shifting with multilayer embedding, *Journal of Visual Communication and Image Representation,* **31** (2015), 64-74. https://doi.org/10.1016/j.jvcir.2015.05.005

[13] J. Tian, Reversible data embedding using a difference expansion, *IEEE Transactions on Circuits and Systems for Video Technology,* **13** (2003), no. 8, 890-896. https://doi.org/10.1109/tcsvt.2003.815962

[14] G. Xuan, Y. Q. Shi, P. Chai, X. Cui, Z. Ni, X. Tong, Optimum histogram pair based image lossless data embedding, Chapter in *Digital Watermarking*, 2008, 264-278. https://doi.org/10.1007/978-3-540-92238-4_21

[15] G. Xuan, C. Yang, Y. Zhen, Y. Q. Shi, Z. Ni, Reversible data hiding using integer wavelet transform and companding technique, Chapter in *Digital Watermarking*, 2005, 115-124. https://doi.org/10.1007/978-3-540-31805-7_10

[16] W. Zhang, B. Chen, N. Yu, Improving various reversible data hiding schemes via optimal codes for binary covers, *IEEE Transactions on Image Processing*, **21** (2012), no. 6, 2991-3003. https://doi.org/10.1109/tip.2012.2187667

[17] The usc-sipi image database, Signal & Image Processing Institute, University of Southern California. http://sipi.usc.edu/database/

Received: January 29, 2017; Published: February 22, 2017